IN THE CLAIMS

1   1. (Previously Amended Once, Currently Amended) A method for managing memory in a

2   computer system, comprising:

3   for at least one memory page,

4   dividing the page into a plurality of relocation blocks, each being

5   identified by an entry of a relocation table; and

6   placing the plurality of relocation blocks at a plurality of locations

7   including one or a plurality of memory systems; and

8   using a relocation table having a plurality of entries to locate the relocation

9   blocks at the plurality of locations;

10   wherein, upon a memory access,

11   using the relocation table to convert an address of the memory page

12   to a relocation address of a relocation block containing the

13   first data intended for the memory access; and

14   if the first data intended for the memory access is not in physical

15   memory, then loading, in physical memory, one or a

16   plurality of relocation blocks containing second data related

17   to the memory access; the second data including at least the

18   first data; and

19   wherein the address of the memory page was converted from a virtual

20   address of the first data.

1   2. (Canceled) The method of claim 1 further comprises the step of converting a virtual

2   address of the data to the address of the memory page.

1   3. (Original) The method of claim 1 further comprises the step of allocating the plurality

2   of relocation blocks corresponding to the memory page upon receiving the address

3   of the memory page.

1   4. (Original) The method of claim 3 further comprises the step of corresponding each

2   entry of the plurality of entries to a particular location of a relocation block.

1   5. (Previously Amended Once, Currently Amended) A system for managing memory in a

2   computer system, comprising:

3   a plurality of relocation blocks located at a plurality of locations including

4   one or a plurality of memory systems; wherein a set of relocation

5   blocks is divided from a memory page; a relocation block being

6   identified by an entry of a relocation table, and an address of the

7   memory page was converted from a virtual address of data in at

8   least one of the relocation blocks; and

9   a the relocation table having a plurality of entries that is used to locate the

10   relocation blocks at the plurality of locations and to convert an the

11   address of the memory page to a relocation address of a relocation

12   block containing the first data intended for a memory access; and

13   if the first data intended for the memory access is not in physical memory,

14   then loading, in physical memory, one or a plurality of relocation

15   blocks containing the second data related to the memory access; the

16   second data including at least the first data.

1   6. (Canceled) The system of claim 5 wherein the address of the memory page was

2   translated from a virtual address of the data.

1 7. (Original) The system of claim 5 further comprises means for allocating the plurality of

2 relocation blocks corresponding to the memory page upon receiving the address of

3 the memory page.

1 8. (Original) The system of claim 7 wherein each entry of the plurality of entries

2 corresponds to a particular location of a relocation block.

1 9. (Previously Amended Once, Currently Amended) A computer-readable medium

2 embodying instructions that cause a computer to perform a method for managing

3 memory in a computer system, the method comprising the steps of:

4 for at least one memory page,

5 dividing the page into a plurality of relocation blocks, each being

6 identified by an entry of a relocation table; and

7 placing the plurality of relocation blocks at a plurality of locations

8 including one or a plurality of memory systems; and

9 using a relocation table having a plurality of entries to locate the relocation

10 blocks at the plurality of locations;

11 wherein, upon a memory access,

12 using the relocation table to convert an address of the memory page

13 to a relocation address of a relocation block containing the

14 first data intended for the memory access; and

15 if the first data intended for the memory access is not in physical

16 memory, then, loading, in physical memory, one or a

17 plurality relocation blocks containing the second data

18 related to the memory access; the second data including at

19 least the first data;

20   wherein the address of the memory page was converted from a virtual

21   address of the data.

1   10. (Canceled) The computer-readable medium of claim 9 wherein the method further

2   comprises the step of converting a virtual address of the data to the address of the

3   memory page.

1   11. (Original) The computer-readable medium of claim 9 wherein the method further

2   comprises the step of allocating the plurality of relocation blocks

3   corresponding to the memory page upon receiving the address of the memory

4   page.

1   12. (Original) The computer-readable medium of claim 11 wherein the method further

2   comprises the step of corresponding each entry of the plurality of entries to a

3   particular location of a relocation block.

1   13. (Currently Added) The method of claim 1 wherein the address of the memory page

2   corresponds to a physical address translated by a translation look-aside buffer.

1   14. (Currently Added) The method of claim 1 wherein an entry of the relocation table

2   includes first address bits corresponding to second address bits maintained by

3   a translation look-aside buffer for use in the memory access.

1   15. (Currently Added) The method of claim 1 wherein if the first data intended for the

2   memory access is not in physical memory, then an address for locating the first

3   data is not in the relocation table.

1   16. (Currently Added) The method of claim 1 wherein the entry corresponding to a

2   relocation block remains pointing to that block when that block moves from

3   one location to another location.

1   17. (Currently Added) The system of claim 5 further comprises a look-aside buffer

2   that maintains address bits for use in the memory access; a first part of the

3   address bits is for use as an offset within a relocation block; a second part of

4   the address bits is for use in indexing into the relocation table.

1   18. (Currently Added) The system of claim 5 wherein if the first data intended for the

2   memory access is not in physical memory, then an address for locating the first

3   data is not in the relocation table.

1   19. (Currently Added) The system of claim 5 wherein the location table is updated

2   after one or the plurality of the relocation blocks is loaded in physical memory.

1   20. (Currently Added) The system of claim 5 wherein the address of the memory page

2   corresponds to a physical address translated by a translation look-aside buffer.

1   21. (Currently Added) A method for managing memory in a computer system,

2   comprising:

3   dividing a memory page into a plurality of relocation blocks each being

4   identified by an entry of a relocation table;

5   upon a request for a piece of data

6   converting a virtual address of the piece of data into a physical

7   address;

8   converting the physical address to an address to identify the

9   piece of data; and

10   if the address identifying the piece of data is in the relocation

11   table, then using an entry of the table to locate the piece

12   of data, else if the address identifying the piece of data

13   is not in the relocation table, then loading, in physical

14   memory, one or a plurality of relocation blocks

15   containing at least the piece of data.

1   22. (Currently Added) The method of claim 21 wherein the piece of data is returned

2   with the physical address in a memory access in response to the request for the

3   piece of data.

1   23. (Currently Added) The method of claim 21 wherein a look-aside buffer maintains

2   address bits for use in a memory access; a first part of the address bits is for

3   use as an offset within a relocation block; a second part of the address bits is

4   for use in indexing into the relocation table.

1   24. (Currently Added) The method of claim 21 wherein a response to the request for

2   the piece of data results in a memory access if the piece of data is not in a

3   cache.

1   25. (Currently Added) The method of claim 21 wherein relocation blocks

2   corresponding to the memory page are created when the memory page is

3   allocated.

1   26. (Currently Added) The method of claim 21 further comprising the steps of

2   allocating a virtual memory page corresponding to a physical address range;

3   translating the physical address range into relocation blocks for each relocation

4   block contained within a physical page, updating an entry corresponding to a

5   relocation block with information to locate that relocation block.

1   27. (Currently Added) The method of claim 21 wherein the entry corresponding to a

2   relocation block remains pointing to that block when that block moves from

3   one location to another location.